

命名数据网络中基于内容请求相关性的协作缓存算法

葛国栋^{*①} 郭云飞^{①②} 刘彩霞^① 兰巨龙^①
^①(国家数字交换系统工程技术研究中心 郑州 450002)
^②(解放军理工大学 南京 210007)

摘要: 针对命名数据网络(Named Data Networking, NDN)存储空间的有效利用和应答内容的高效缓存问题, 该文采用“差异化缓存”的方式, 提出一种依据内容请求序列相关性的协作缓存算法。在内容请求中, 预先发送对于后续相关数据单元的并行预测请求, 增大内容请求的就近响应概率; 缓存决策时, 提出联合空间存储位置与缓存驻留时间的 2 维差异化缓存策略。根据内容活跃度的变化趋势, 空间维度上逐跳推进内容存储位置, 时间维度上动态调整内容缓存时间, 以渐进式的方式将真正流行的请求内容推送至网络边缘存储。该算法减小了内容请求时延和缓存冗余, 提高了缓存命中率, 仿真结果验证了其有效性。

关键词: 互联网; 命名数据网络(NDN); 协作缓存; 请求相关性; 内容路由

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2014)12-2795-07

DOI: 10.3724/SP.J.1146.2014.00114

Collaborative Caching Algorithm Based on Request Correlation in Named Data Networking

Ge Guo-dong^① Guo Yun-fei^{①②} Liu Cai-xia^① Lan Ju-long^①

^①(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

^②(People's Liberation Army University of Science and Technology, Nanjing 210007, China)

Abstract: How to efficiently utilize the finite storage space and cache content chunks in the content store poses challenges to the caching policy in Named Data Networking (NDN). Using the differentiated caching strategy, a collaborative caching algorithm is proposed based on the request correlation. In the scheme, the subsequent correlated content chunks are requested in advance to increase the hit ratio for content requesting. When making the caching decision, a two-dimensional differentiated caching policy combining the caching location and cache-resident time is proposed. According to the change of content activity, the caching location is pushed downstream hop by hop in the spatial dimension in order to spread popular contents to the network edge in a gradual manner, and the cache-resident time is adjusted dynamically in the time dimension. The simulation results show that the proposed algorithm can efficiently decrease the request latency, reduce the cache redundancy, and achieve higher cache hit ratio than other caching strategies.

Key words: Internet; Named Date Networking (NDN); Collaborative caching; Request correlation; Content-based routing

1 引言

随着互联网技术与应用的飞速发展,“宽带化”、“内容化”与“个性化”已经成为网络发展的主旋律,人们对于数据内容的需求日益强烈,网络应用的主体逐步向内容请求和信息服务演进转移^[1]。据 Cisco VNI Mobile Forecast 预测,到 2014 年互联网上所有内容相关的流量将占据超过 97.5%的份额,

传统以主机为中心的网络体系结构难以满足当前网络信息服务的发展要求。为此,信息中心网络(Information-Centric Networking, ICN)^[1]作为一种革命式(clean-slate)的未来互联网设计思路,让数据内容本身成为网络通信的主体单元,将网络通信模式从关注“在哪”(地址、服务器)转变为关注“是什么”,即用户和应用通信的目的和意向,成为未来 Internet 设计的重要模式。其中,命名数据网络(Named Data Networking, NDN)^[2]作为典型的 ICN 网络体系结构范例,在中间层用命名数据取代 IP,数据传输采用“发布-请求-响应”模式,直接以内容

2014-01-20 收到, 2014-04-24 改回

国家 973 计划基金(2012CB315901), 国家自然科学基金(61372121), 和国家 863 计划项目(2011AA01A103)资助课题

*通信作者: 葛国栋 ggd@mail.ndsc.com.cn

名字(named data)进行路由, 实现点到多点的高效内容分发。

在 NDN 的设计中, 采用网络内在普遍缓存(in-network caching)的方式, 在兴趣包(interest packet)沿途转发路径(on-path)的所有节点上缓存应答内容。在路由转发中, 当节点收到兴趣包, 依据内容名字依次在内容存储器(Content Store, CS), 未决请求表(Pending Interest Table, PIT)和转发信息库(Forwarding Information Base, FIB)中进行匹配查询。应答数据包(data packet)携带请求内容, 依据节点 PIT 表项的记录, 沿相同路径进行反向的逐跳传输。

文献[3]指出, 对于 NDN 网络, 合理的内容放置和缓存决策, 是有效发挥网络性能的关键因素。但由于 NDN 泛滥式的沿途全部缓存方式(Cache Everything Everywhere, CE²), 致使节点缓存内容趋于同质化, 导致大量的缓存冗余。而且, 在缓存决策时, 缺乏对于内容本身差异化特征的考虑, 无法实现内容的优化存储。为此, 为了有效发挥内容普遍缓存的优势, 高效缓存算法的设计就成为 NDN 需要解决的关键问题。

2 问题分析

文献[4]提出了一种随机单点缓存策略(RCOne), 在沿途路径上随机选择单个节点进行内容存储, 减小缓存冗余。但该方案没有考虑不同请求内容流行程度的差异性; 文献[5]提出只在缓存命中节点的直接下一跳存储应答内容(LCD), 避免内容的重复缓存; 文献[6]提出了基于节点介数的缓存方案(betw), 通过在沿途传输路径介数最大的节点上缓存内容, 以提高后续利用率。但该方案将导致少数重要节点上的缓存频繁替换; 文献[7]设计了基于概率的缓存方式(probabilistic cache), 依据节点距离数据源的距离和路径的剩余存储能力, 计算节点对应的缓存概率; 文献[8]提出了缓存年龄的思想(age), 依据内容流行等级和存储节点的位置来计算缓存年龄。但是该算法中假设内容流行度是已知的静态参量, 无法体现内容请求的动态变化特性, 而且该参数难以预先获取; 文献[9]提出了一种基于内容流行等级的协作缓存策略(WAVE), 依据内容的请求次数, 以指数增长方式逐步增加沿途缓存的数据个数。但是该方案并没有考虑内容请求序列的相关性, 而且只实现了空间存储位置上的差异化缓存; 文献[10]对现有缓存方案进行了对比分析, 指出缓存算法的设计缺乏对于内容请求分布特征的考虑, 并给出下一步的研究方向。

以上方案存在的不足之处主要体现在: (1)在差异化缓存设计时, 对于不同内容的存储位置和缓存驻留时间缺乏综合考虑和有效结合; (2)缓存决策没有结合内容请求序列的分布特征。缓存算法设计时, 多数基于独立请求模式(Independent Reference Model, IRM)进行研究^[10], 内容请求概率由流行度分布函数预先给定, 无法体现请求分布的动态变化和相关性特征^[10, 11]。针对上述不足, 在 NDN 中本文提出了一种基于内容请求相关性的协作缓存算法(Collaborative Caching Algorithm based on Request Correlation, CCARC)。CCARC 根据内容请求序列间的相关性特征, 主动发送对于关联数据单元的并行预测请求; 在缓存决策时, 从差异化缓存思想出发, 根据内容活跃度的变化趋势, 逐步推进内容存储位置, 以渐进式的方式将流行资源推送至网络边缘节点存储。

3 基于内容请求相关性的协作缓存算法

CCARC 属于一种隐式的轻量级协作缓存算法, 节点以分布式的方式独立地执行缓存决策, 不会引入额外的交互报文的发送与通告, 也避免了中心化决策实体的引入。同时, 内容的流行程度根据实际请求次数来动态确定, 不依赖预先的参数假设。CCARC 主要思想包括两个方面: (1)基于内容请求相关性的并行预测请求; (2)缓存位置和驻留时间的 2 维差异化存储。

3.1 报文交互与参数定义

为了区分基于相关性预测发起的请求和应答数据, 分别设计了相关兴趣包(Correlated Interest packet, CIP)和相关数据包(Correlated Data Packet, CDP)报文, 其具体格式与 NDN 中的兴趣包和数据包相同, 只是 CIP 和 CDP 用于标识节点基于请求相关性而发出的交互报文, 用于执行下一相关数据单元(chunk)的预测请求。在数据包和 CDP 报文中, 添加缓存指示(Cache Indication, CI)和内容活跃度(Content Activity, CA)字段。其中, CI 用于缓存节点指示下游节点执行应答内容的存储, CA 表示该内容对应的流行程度, 用于存储位置和缓存时间(Caching Time, CT)的计算, 图 1 给出了具体报文格式。

定义 1 内容请求相关系数 ρ_{cor} : NDN 中是基于细粒度的 chunk 单元执行内容缓存和请求的, 连续请求的多个 chunk 很可能属于同一内容对象^[10, 11]。 ρ_{cor} 用于定量地描述内容请求序列之间的相关性程度, 表示同一内容对象, 相邻 chunk 单元被连续请求的概率大小:

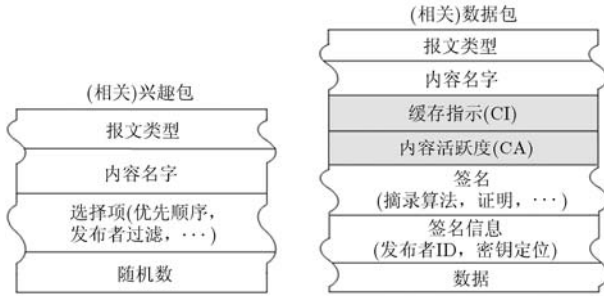


图 1 (相关)兴趣和(相关)数据报文格式

$$\rho_{cor} = P(r_{n+1} = C_{i+1} | r_n = C_i) \quad (1)$$

其中, $0 \leq \rho_{cor} \leq 1$, r_{n+1} , r_n 分别表示第 n 和 $n+1$ 次发送的内容请求, $\forall i, C_i$ 和 C_{i+1} 属于同一内容对象连续的数据块单元。 ρ_{cor} 越大, 请求序列之间的相关性越强, CIP 报文的预测内容命中率越高。

定义 2 内容活跃度(Content Activity, CA): CA 定义为时间间隔 T 中内容被请求的次数。为了体现内容请求分布的动态变化和局域特性, 采用滑动时间窗口机制, 对 CA 进行动态地计算与更新, 兼顾内容对象历史请求的“热度”和当前时刻的“新颖度”。图 2 给出了 CA 计算更新的图示。

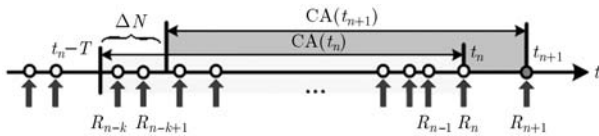


图 2 内容活跃度 CA

当沿途缓存节点接收到内容 C 的第 $n+1$ 次请求 r_{n+1} 时, 对应的内容活跃度 $CA(t_{n+1})$ 更新为

$$CA(t_{n+1}) = CA(t_n) + 1 - \Delta N \quad (2)$$

其中, t_n, t_{n+1} 为内容请求的到达时刻, T 为统计时间间隔, ΔN 表示在 $[t_n - T, t_{n+1} - T]$ 区间中内容请求的到达次数。在缓存决策时, 将依据 CA 的动态变化, 确定内容的存储位置并计算相应的缓存时间 CT。

3.2 算法执行流程

3.2.1 缓存存储决策 节点依据接收到的报文格式和内容活跃度 CA 的变化趋势, 动态地确定内容存储位置, 计算、更新缓存时间。具体执行以下 3 种策略:

(1)当缓存节点接收到兴趣包时, 说明该内容为终端用户所请求, 节点计算内容活跃度 CA。若 $CA(t_{n+1}) \geq CA(t_n)$, 将缓存标志 CI 置为 1, 指示下行节点缓存内容。下游节点接收到数据包后, 查看缓存标志位, 如果 CI 为 1, 说明该节点为目标缓存

节点, 需要存储内容并依据 CA 计算对应的缓存时间 CT。在 CS 中缓存内容后, 将数据包中的缓存标志位重置为 0, 避免下游节点的重复冗余存储;

(2)若 $CA(t_{n+1}) < CA(t_n)$, 说明当前内容活跃程度降低, 将 CI 置为 0, 停止该内容缓存位置的下行推进, 依据 CA 取值直接在该节点调整内容缓存时间 CT, 并发送数据包进行响应。下游节点接收到应答报文后, 由于 CI 为 0, 无需进行内容存储, 依据 FIB 直接进行报文转发;

(3)当缓存节点接收到 CIP 报文时, 说明该数据单元为下游节点基于相关性预测发起的内容请求, 节点发送 CDP 报文进行响应。当请求节点接收到 CDP 报文后, 提取应答内容, 以临时缓存时间 T_{tem} 在 CS 中进行存储, 用于后续内容请求的就近应答。

3.2.2 缓存时间计算 (1)缓存时间 CT 的确定。CT 取值依据 CA 来动态计算, 内容活跃度越高, 对应的缓存时间越长, 以增大流行内容的驻留时间和后续利用率。若 $CA(t_{n+1}) < CA(t_n)$, 则直接在该节点依据 CA 取值更新 CT。当 $CA(t_{n+1}) \geq CA(t_n)$, 将数据包中的 CI 置为 1, 并携带 CA 取值, 指示下行节点缓存内容并进行 CT 的计算。如果缓存节点接收到的是相关性预测请求报文 CIP, 由于该内容请求并不是用户发起的, CA 字段保持不变, 无需进行 CT 的计算更新。

当内容首次请求(CA=1)时, 由于沿途节点不存在就近的缓存资源, 内容请求需要发送至源服务器进行响应, 应答内容将在源服务器的直接下一跳节点进行存储, 对应的缓存时间设定为基础缓存时长 T_{base} , $CT(1) = T_{base}$ 。后续内容请求对应的缓存时间 CT 在 T_{base} 基础上依据 CA 取值进行动态地计算更新:

$$CT(n+1) = CT(n) + \left(1 - \frac{1}{n+1}\right) \cdot T_{base}, \quad CA = n+1 \quad (3)$$

其中, $n, n+1$ 表示时间 T 中内容请求到达次数, T_{base} 的确定可参考文献[8]的设定。

(2)临时缓存时间 T_{tem} 。由于临时缓存的目的是基于用户会在短时间内发起下一个相关数据单元的请求, 从而预先进行内容的获取, 减小后续内容请求时延。所以, T_{tem} 的取值应依据内容请求的平均发送时间间隔来进行设定, 而不依赖于内容的实际请求次数和活跃程度。

3.2.3 缓存替换策略 当节点 CS 存储单元已全部利用, 没有可用的缓存空间时, 需要删除已有缓存内容, 用于最新应答内容的存储。CCARC 中, 缓存时间到期的内容将被标记为“可替换”状态, 当需要执行缓存替换时, 优先删除“可替换”数据单元。

如果没有过期内容, 将替换 CS 中缓存时间 CT 最小的 chunk 用于最新到达内容的存储。在 CCARC 中, 在缓存决策时, 依据内容的 CA 来计算 CT, 流行程度越高的内容将被赋予更大的缓存时长。同时, 对于基于相关性预测请求的内容执行临时缓存, 其对应的存储时间 T_{tem} 远远小于正常基于 CA 计算的 CT。如果用户后续没有发送对于该预测内容的请求, 临时缓存内容将很快衰减为“可替换”状态, 被更新淘汰。

3.3 算法运行实例

图 3 给出了 CCARC 算法具体执行实例。假设内容 C 包括 7 个 chunk ($C_1 \sim C_7$), 存储于内容源服务器 (Original Content Server, OCS)。终端用户 $U_1 \sim U_4$ 为内容请求者, A, B, C, D 均为内容路由器 (Content Router, C-Router), 实线方框表示正常的缓存单元, 虚线方框代表临时存储的数据单元。

(1) U_1 发送对前 5 项数据块 ($C_1 \sim C_5$) 的请求 (图 3(a)): 由于沿途路径不存在对应的缓存资源, 兴趣包请求需要路由至 OCS 进行响应。当 OCS 接收到内容请求后, 发送数据包进行应答, 并将缓存指示标志 CI 位置为 1, 指示下游路由器缓存该应答内容; 当 C-Router A 接收到数据包后, 查看 CI 标志和

CA 字段, 计算缓存存储时间 CT, 将内容在其 CS 中进行存储, 并重置 CI 为 0, 继续向下转发应答数据包。当 B, C 和 D 接收到数据包后, 由于 CI 标志位为 0, 直接进行报文转发, 无需进行数据缓存;

(2) U_2 发送对前 3 项数据内容 ($C_1 \sim C_3$) 的请求 (图 3(b1)): 在首次请求中, 由于 $C_1 \sim C_5$ 在路由器 A 处已经进行了缓存, 节点 A 直接发送 $C_1 \sim C_3$ 对应的数据包进行应答, 将 CI 置为 1, 指示下行路由器 B 缓存该应答内容。 $C_1 \sim C_3$ 将推送至节点 B 处进行存储; 假设在该次请求中 (图 3(b2)), C_3 对应的内容活跃度 CA 降低, 那么对应的数据包中 CI 将设置为 0, 缓存位置将不会向下 (节点 B) 推进, 直接在节点 A 处依据当前 CA 重新计算缓存时间 CT;

(3) U_3 发送对前 4 项数据块 ($C_1 \sim C_4$) 的请求 (图 3(c)): 由于沿途缓存资源的存在, $C_1 \sim C_3$ 对应的兴趣包将在 B 处得到响应, 并在 C 处依次进行缓存。当节点 B 进行 C_3 的应答时, 同时发送 CIP 报文, 执行对于数据块 C_4 的相关性预测请求。当上游节点 A 接收到 CIP 后, 发送 CDP 进行应答。当节点 B 接收到 C_4 的 CDP 应答报文后, 提取内容并进行临时存储 (虚线方框)。当 C-Router B 接收到用户发送的 C_4 请求兴趣包后, 该内容将被节点正式缓存 (实线方框), 并依据 CA 重新计算缓存时间 CT。当 C_5 在节点 B 进行临时缓存后, 由于用户没有发起对于该内容的请求, 该内容短时间内将会被替换淘汰。

CCARC 算法的整体流程示于表 1, 划分为内容请求和数据应答两个过程。

4 仿真与性能分析

4.1 仿真环境与参数设置

采用 ndnSIM^[12] 进行仿真与性能分析, 该工具对于 NDN 的基本数据单元结构和路由转发流程均已实现, 并提供了开放的源码和运行实例。在 GT-ITM 下采用 Locality 模型生成 50 个路由节点的平面随机网络拓扑。网络中内容对象总数 N 为 3000 个, 以 1~3000 依次排序, 每个内容划分为 100 个 chunk^[13], 大小设为 10 kB。节点缓存容量一致, CS 设为 200 MB, 链路带宽 100 Mbps。在网络中设置 3 个内容服务器 OCS, 负责内容对象的存储和发布, 各 OCS 随机存储 1000 个内容。其余节点作为用户接入节点, 内容请求到达服从参数 λ 的泊松过程, 请求概率服从 Zipf 分布^[13], 并依据相关性系数 ρ_{cor} 依次发送对于后续数据单元的请求。为了构造内容请求分布的动态变化, 在内容请求对应 Zipf 分布序列中随机选取一定比例的扰动内容, 进行随机化重新排序^[14], 扰动比例选取 5%。仿真时间设为 200 s, $T_{base} = 10$ s^[8], 临时缓存时间 T_{tem} 设为 2 倍的内容发送间隔 200 ms, 采样周期 $T = 2$ s。初始节点缓存状态为空, 缓存替换采用最近最少使用策略 LRU。

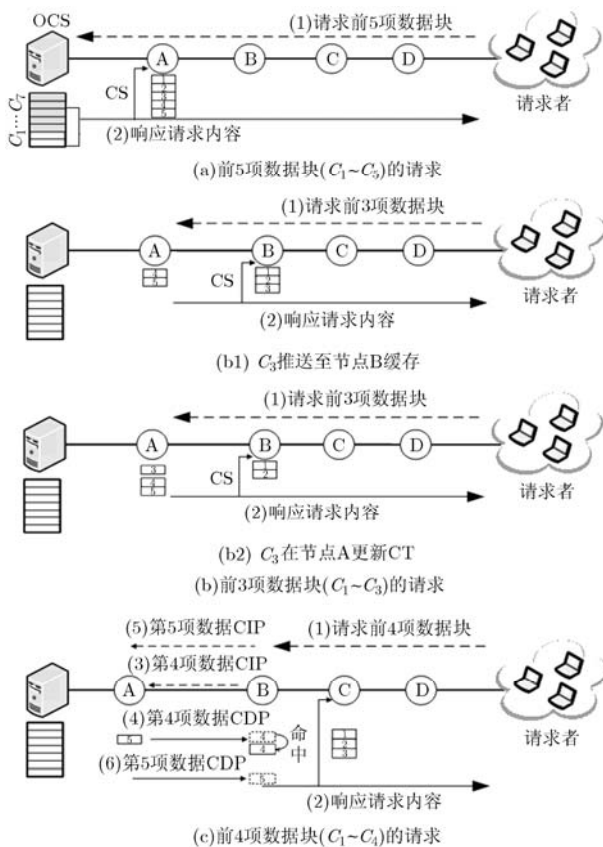


图 3 CCARC 算法执行过程

表 1 CCARC 算法的整体流程

初始化: Content Name(C_i), CI, RT
内容请求过程: 发送内容请求(兴趣包或 CIP)
1 foreach(C-Router from Content Consumer to Content Server)
2 if CS 包含请求内容 C_i then
3 if 接收到兴趣包报文 then
4 计算 $CA(t_{n+1})$;
5 if $CA(t_{n+1}) < CA(t_n)$ then
6 依据 $CA(t_{n+1})$ 更新缓存时间 CT;
7 else 设置缓存标志位 $CI=1$, 指示下行节点缓存;
8 发送数据包进行响应;
9 发送 CIP, 执行对于 C_{i+1} 相关性预测请求;
10 if 接收到相关兴趣包报文 then
11 发送相关数据包进行应答;
12 else 按照 FIB 直接进行兴趣包转发;
数据响应过程: 接收内容应答(数据包或 CDP)
13 foreach(downstream C-Router)
14 查看缓存标志位 CI;
15 if 接收到数据包 and $CI=1$ then
16 依据 $CA(t_{n+1})$ 计算缓存时间 CT;
17 if CS 空间已满 then
18 执行缓存替换更新;
19 在 CS 中缓存内容 C_i ;
20 重置 $CI=0$, 继续下行转发数据包;
21 if 接收到相关数据包 and $CI=1$ then
22 if CS 空间已满 then
23 执行缓存替换更新;
24 在 CS 中临时缓存内容 C_{i+1} , $CT=T_{tem}$;
25 if 接收到兴趣包 then
26 计算 $CA(t_{n+1})$ 和缓存时间 CT;
27 在 CS 中正式缓存内容 C_{i+1} ;

4.2 性能分析

将 CCARC 与文献[2] NDN, 文献[4]RCOne 和文献[9] WAVE 进行对比分析, 评价指标包括: 平均请求时延(Average Request Delay, ARD), 缓存命中率(Cache Hit Ratio, CHR), 平均路由跳数(Average Route Hop, ARH)。

(1)平均请求时延 平均请求时延 ARD: 节点发送请求兴趣包到接收到应答数据包之间的平均延迟。图 4 给出了 $\alpha=1.0$, $\rho_{cor}=0.95$, λ 分别取值 10 和 20 时, 4 种方案的 ARD 对比。仿真初始阶段, 由于节点存储状态为空, 发送的兴趣包请求都需要转发至 OCS 进行响应, 请求时延较大。但随着内容的不断存储, 缓存内容的响应概率逐渐增加, ARD 随之减小。

由于 NDN 的 CE^2 泛滥式缓存方式, 节点的同质化缓存将会导致高频率的内容替换更新, 降低了沿途缓存资源的响应率, 对应的 ARD 最大; RCOne 在沿途路径上以固定概率随机选择单个节点进行内容存储, 但该方案没有考虑不同请求内容流行程度的差异性, 固定的缓存概率无法实现内容的优化存储; WAVE 依据内容的请求次数, 以指数增长方式不断增加传输路径缓存数据个数。但是该方案并没有考虑内容请求分布的相关性特征。相比其他方案, CCARC 对于不同请求内容, 执行差异化缓存。依据内容活跃度的变化趋势, 计算内容存储位置, 动态调整内容缓存时间, 增大了流行资源驻留概率和命中率。同时, 依据内容请求序列的强相关性, 主动发送对于关联内容的预测请求, 有效减小内容请求时延(约 37%)。

(2)缓存命中率 缓存命中率 CHR: 网络中节点缓存内容响应兴趣包请求的概率。缓存内容的命中率越高, 对应的源服务器响应率和负载就越小。图 5 给出了 $\alpha=1.0$, $\rho_{cor}=0.95$, λ 取值 10, 20 时, 4

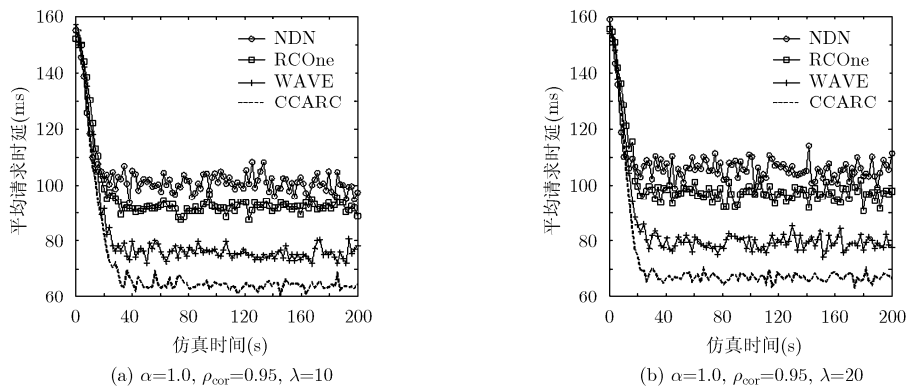


图 4 平均请求时延 ARD 对比

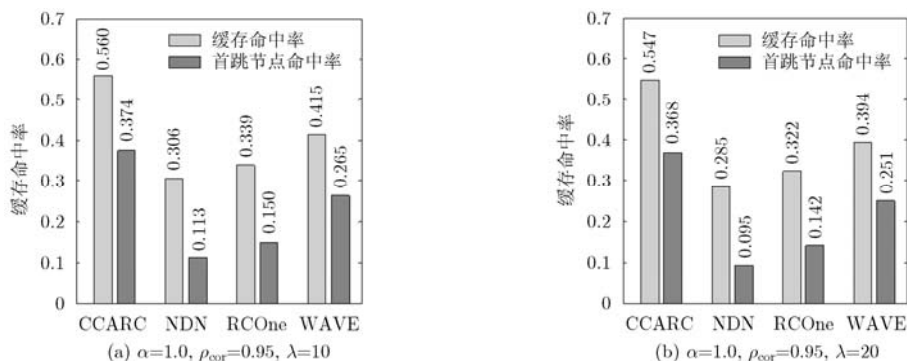


图 5 缓存命中率 CHR 对比

种方案的 CHR 对比, 其中深色部分表示首跳节点命中率。对于 NDN 和 RCOne, 在应答内容存储时, 没有结合不同请求内容流行等级的差异性, 致使节点缓存内容的频繁替换, 增大了缓存缺失概率, CHR 明显小于 CCARC。WAVE 在缓存决策时, 对于不同流程度度的请求内容, 只实现了空间存储位置的差异化缓存, 对于时间上内容的驻留时长缺乏考虑。CCARC 在缓存决策时, 综合考虑请求内容在空间和时间维度上的差异化存储, 增大沿途节点缓存命中率, CHR 分别达到了 0.560 和 0.547。而且, 内容流行程度越高, 缓存位置越靠近网络边缘节点, 增大了内容请求的就近响应率, 首跳命中率分别达到了 0.374 和 0.368。

(3) 平均路由跳数 平均路由跳数 ARH: 定义为兴趣包命中请求内容时传输的平均距离, 以路由跳数计算。图 6 给出了 $\alpha=1.0, \rho_{cor}=0.95, \lambda=10$ 时, 各方案 ARH 随节点缓存容量 CS 大小的变化趋势。当沿途节点 CS 空间为 0 时, 内容请求都需要发送至 OCS 进行响应, 各方案对应的 ARH 无明显差别。随着节点缓存容量的增加, 更多的请求内容可以被存储在传输节点上, 提高了沿途路径缓存响应率, 各方案对应的 ARH 不断减小。CCARC 依据内容请求次数, 将节点有限的存储空间用于缓存活跃度最高的请求内容, 应答内容只在命中节点直接的下一跳进行存储, 减小相同内容副本重复存储的冗余。同时, 依据数据序列的强相关性, 主动发起对于相关数据单元的并行预测请求, 减小了后续内容请求的路由传输距离。在有限的缓存容量下, CCARC 可以充分利用已有的缓存资源, ARH 是各方案中最小的。

4.3 适应性讨论

图 7 给出了在 $\rho_{cor}=0.95$ 下, ARD 随 Zipf 指数 α 变化的趋势。当 α 取值较小时, 内容请求分布不能有效集中, 多样化的内容请求和存储将导致高频率的缓存替换更新, 增大了内容请求时延。随着 α 参数的增大, 内容请求的集中性和局域性不断加强,

流行资源在缓存中的驻留概率和响应率明显增大, 请求时延逐步减小。特别是当 α 取值位于 0.8 到 1.2 区间时, 内容请求的集中性急剧增大, 80% 的内容请求包含的内容对象数量从 1223 项集中至 118 项, ARD 大幅下降。随着 α 取值的进一步增大 ($\alpha > 1.2$), 流行资源对应的分布已无显著变化, ARD 下降趋势逐渐减缓。CCARC 对于不同流行度的请求内容执行差异化缓存, 并通过相关性预测请求, 预先获取内容, 有效地降低了 ARD, 对于 α 取值变化具有良好的适应性。

图 8 给出了在相关系数 ρ_{cor} 取值为 1.0, 0.9 和 0.8 下, ARD 随请求内容所含 chunk 个数 (Number of Chunks, NoC) 的变化趋势。当 NoC=1 时, 每个内容只含有一个数据块单元, 对于 CCARC, 无法执行后续相关数据单元的预测请求, 在不同的 ρ_{cor} 取值下, ARD 几乎不变。随着请求内容所含 chunks 个数不断增大, CCARC 执行后续相关数据单元的预测请求愈加充分, 内容在沿途缓存资源处响应的概率越高, ARD 不断减小。同时, 在不同的相关性指数 ρ_{cor} 取值下, CCARC 可以有效降低 ARD, ρ_{cor} 取值越大, 内容请求序列相关性程度越强, CIP 报文请求的预测内容命中率越高, 对应的 ARD 越小。

5 结束语

内容的普遍缓存是 NDN 网络的核心特征, 而合理的内容放置和缓存策略是其性能有效发挥的保证。本文针对现有缓存算法的不足, 结合内容请求分布的相关性特征, 从差异化缓存的思想出发, 在 NDN 中提出了一种依据内容请求序列相关性的协作缓存算法。在内容请求过程中, 节点主动发送对于下一个相关数据单元的预测请求, 增大后续内容请求的就近应答; 在缓存决策时, 逐跳推进内容存储位置, 动态调整缓存驻留时间, 实现不同内容的差异化存储。后续研究工作包括: (1) 如何将 CCARC 缓存算法扩展到多数据源和多路径模式下; (2) 在不同网络和仿真参数条件下对于 CCARC 性能进一步分析验证。

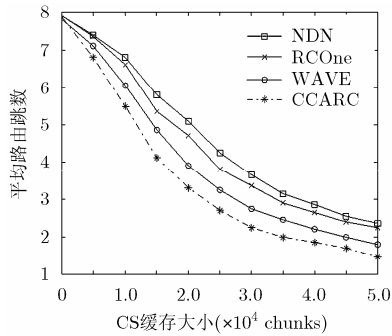


图 6 平均路由跳数 ARH 对比

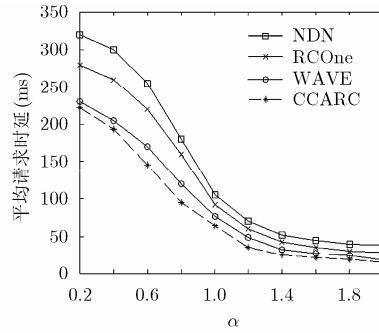
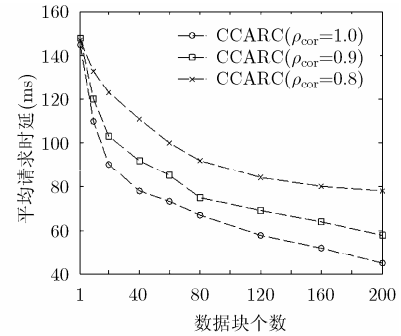
图 7 平均请求时延 ARD 随 Zipf 指数 α 的变化趋势

图 8 平均请求时延 ARD 随 NoC 的变化趋势

参考文献

- [1] Ahlgren B, Dannewitz C, Imbrenda C, *et al.* A survey of information-centric networking[J]. *IEEE Communication Magazine*, 2012, 50(7): 26-36.
 - [2] Jacobson V, Smetters D K, Thornton J D, *et al.* Networking named content[J]. *Communications of the ACM*, 2012, 55(1): 117-124.
 - [3] Chiocchetti R, Perino D, Carofiglio G, *et al.* INFORM: a dynamic Interest Forwarding Mechanism for Information Centric Networking[C]. ACM SIGCOMM Workshop on Information-Centric Networking, Hong Kong, China, 2013: 9-14.
 - [4] Eum S, Nakauchi K, Murata M, *et al.* CATT: potential based routing with content caching for ICN[C]. ACM SIGCOMM Workshop on Information-Centric Networking, Helsinki, Finland, 2012: 49-54.
 - [5] Laoutaris N, Syntila S, and Stavrakakis I. Meta algorithms for hierarchical web caches[C]. IEEE International Conference on Performance, Computing, and Communications, Phoenix, USA, 2004: 445-452.
 - [6] Chai W K, He D, Psaras I, *et al.* Cache "less for more" in information-centric networks[C]. Proceedings of IFIP 6 Networking Conference, Prague, Czech, 2012: 27-40.
 - [7] Psaras I, Chai W K, and Pavlou G. Probabilistic in-network caching for information-centric networks[C]. ACM SIGCOMM Workshop on Information-Centric Networking, Helsinki, Finland, 2012: 55-60.
 - [8] Ming Z X, Xu M W, and Wang D. Age-Based cooperative caching in information-centric networks[C]. IEEE INFOCOM, Orlando, USA, 2012: 268-273.
 - [9] Cho K, Lee M, Park K, *et al.* WAVE: popularity-based and collaborative in-network caching for content-oriented Networks[C], IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking, Orlando, USA, 2012: 316-321.
 - [10] Zhang G Q, Li Y, and Lin T. Caching in information centric networking: a survey[J]. *Computer Networks*, 2013, 57(16), 3128-3141.
 - [11] 朱轶, 糜正琨, 王文鼎. 一种基于内容流行度的内容中心网络缓存概率置换策略[J]. *电子与信息学报*, 2013, 35(6): 1305-1310.
Zhu Yi, Mi Zheng-kun, and Wang Wen-nai. A cache probability replacement policy based on content popularity in content centric networks[J]. *Journal of Electronics & Information Technology*, 2013, 35(6): 1305-1310.
 - [12] NS-3 based Named Data Networking (NDN) Simulator[OL]. <http://ndnsim.net>. 2013.
 - [13] Chiocchetti R, Rossi D, Rossini G, *et al.* Exploit the known or explore the unknown? Hamlet-like doubts in ICN[C]. ACM SIGCOMM Workshop on Information-Centric Networking, Helsinki, Finland, 2012: 7-12.
 - [14] Guo S, Xie H Y, and Shi G. Collaborative forwarding and caching in content centric networks[C]. Proceedings of IFIP Networking, Prague, Czech Republic, 2012: 41-55.
- 葛国栋：男，1985年生，博士生，研究方向为新型网络体系结构设计、内容中心网络。
- 郭云飞：男，1963年生，硕士，教授，博士生导师，研究方向为新型网络体系结构设计、移动互联网。
- 兰巨龙：男，1962年生，博士，教授、博士生导师，研究方向为可重构柔性网络和高性能路由。
- 刘彩霞：女，1974年生，博士，副教授，硕士生导师，研究方向为内容中心网络、移动互联网。